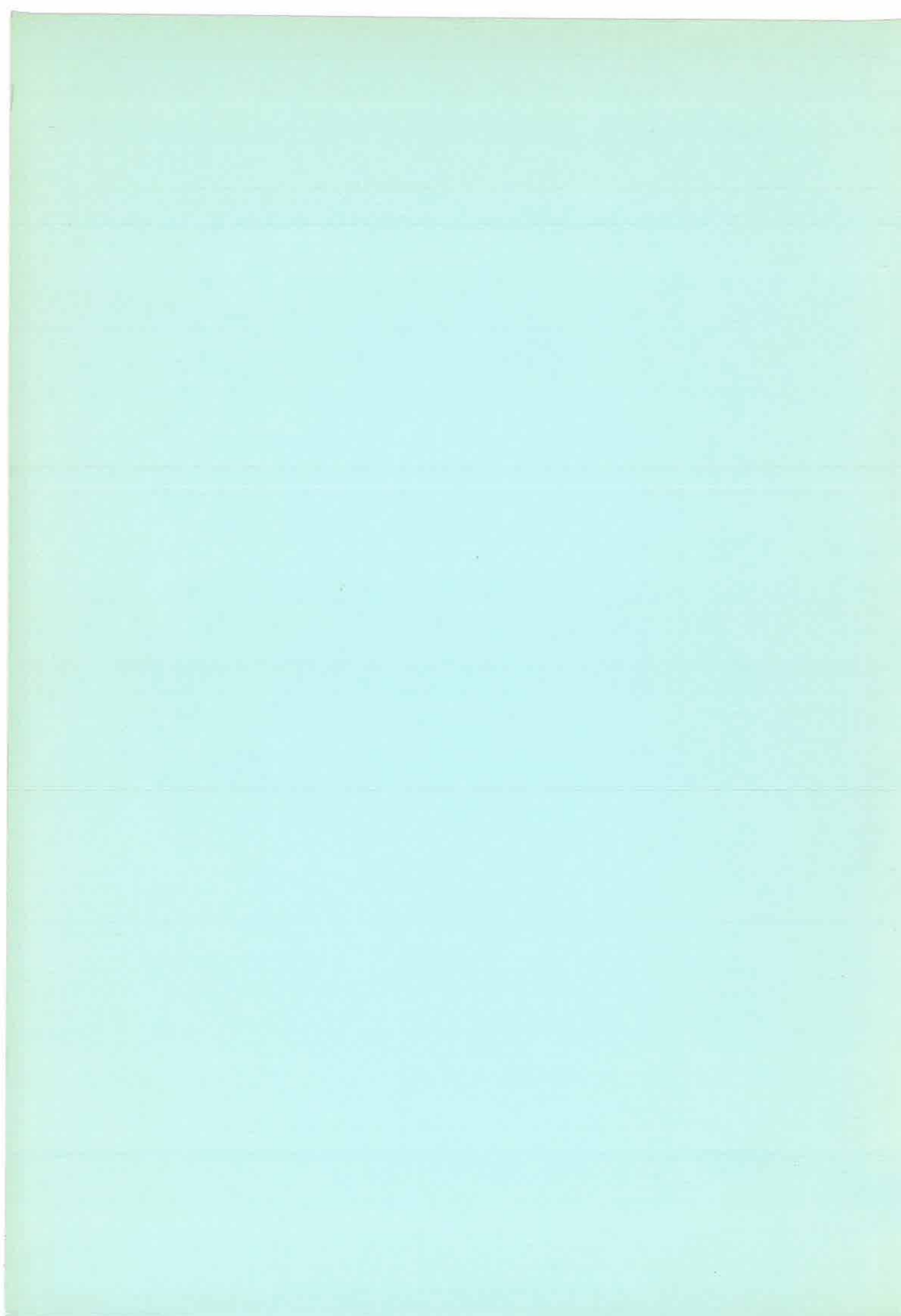


Het groeperen van frequentieverdelingen

H.H.F.M. Verstralen



OPD Memorandum

98-3

Het groeperen van frequentieverdelingen

H.H.F.M. Verstralen

Cito
Arnhem, 1998

Cito Instituut voor Toetsontwikkeling
Postbus 1034 6801 MG Arnhem
Bibliotheek

8501 001 6425



Prijs f 15,--

© Cito Arnhem 1998

Uit deze uitgave mogen zonder toestemming van de auteur geen aanhalingen worden overgenomen.

Voorwoord

Voor het toepassen van sommige statistische technieken moet een frequentieverdeling worden samengenomen tot een klein aantal groepen van zoveel mogelijk gelijke grootte. Het referentiepunt is daarom de groepering met groepen van gelijke grootte, en een optimale groepering moet daar zoveel mogelijk op lijken. Een optimale groepering is niet eenvoudig te vinden, zodat voor het opsporen een algoritme nodig is. In het voorliggende rapport worden enkele eigenschappen van groeperingen besproken. Tevens worden enkele algoritmen voor het zoeken naar een optimale groepering ontwikkeld en geëvalueerd.

Steekwoorden: frequentieverdelingen, homogeen groeperen.

Huub Verstralen

Arnhem, maart 1998

Inhoudsopgave

Voorwoord

1	Inleiding	1
2	Kenmerken van groeperingen	3
3	Deterministische algoritmen voor het vinden van een homogene groepering	11
4	Een simulatiestudie	17
5	Een random algoritme	23
6	Conclusie	25
	Appendix	27

1 Inleiding

Voor het beoordelen van de kwaliteit van een meerkeuzevraag is het handig om de relatie tussen toetsscore en bijvoorbeeld de kans op een goed antwoord in beeld te brengen. Hetzelfde geldt voor het berekenen van een toetsingsgrootte waarmee per score een door het model voorspelde statistiek wordt vergeleken met de geobserveerde waarde daarvan. Vaak heeft men echter te weinig observaties per score om bijvoorbeeld de kans op goed, gegeven de score, voldoende betrouwbaar te schatten. In een dergelijk geval moeten de scores worden gegroepeerd. Over het algemeen is het dan gewenst, dat de aantallen observaties in de opeenvolgende groepen zoveel mogelijk gelijk zijn. Een dergelijke groepering noemen we homogeen. In deze notitie worden kenmerken van homogene groeperingen onderzocht. Vervolgens worden algoritmen besproken om homogene groeperingen te vinden.

2 Kenmerken van groeperingen

Het aantal observaties in een groep noemen we de grootte van een groep. Voor het onderling vergelijken van groeperingen onderling op hun mate van homogeniteit, moet daarvoor een kwantitatieve maat worden gekozen. Men zou daarvoor bijvoorbeeld de variantie van de groepsgroottes kunnen nemen. In dit rapport kiezen we het verschil tussen de grootste en de kleinste groep van een groepering. Dit verschil noemen we de range. Een optimale groepering heeft de kleinste range van alle mogelijke groeperingen. De grootte van de kleinste groep(en) noemen we het minimum van de groepering. Dat niet alle groeperingen van een frequentieverdeling met een minimale range hetzelfde minimum hoeven te hebben, kan eenvoudig geïllustreerd worden met de volgende twee groeperingen van dezelfde frequentieverdeling

Groepering1	2		5		5		
Frequenties	2		4		1	2	3
Groepering2		6			3		3

De scheidingsstreepjes geven aan welke frequenties bij elkaar worden genomen, en de getallen ernaast of ertussen zijn gelijk aan de som van de gegroepeerde frequenties. In beide groeperingen is de range gelijk aan 3, maar de kleinste groep van Groepering1 telt 2 observaties, terwijl die van Groepering2 er een meer bevat. Omdat een statistiek van een groep nauwkeuriger wordt geschat naarmate zij groter is, noemen we een groepering pas dan optimaal wanneer zij niet alleen de kleinste range maar ook de grootste minimum groep heeft.

Behalve op de range van een groepering als geheel, kan er ook worden gelet op de range van een deel daarvan. Neem bijvoorbeeld de volgende frequentieverdeling en Groepering1 in 4 groepen.

Frequenties	32	29		300		6	55		7	74
Groepering1	61		300		61		81			

Groepering1 is optimaal met $\text{range} = 300 - 61 = 239$. De volgende groepering

Frequenties	32	29	300	6	55	7	74
Groepering2	61		300		74		74

heeft dezelfde range en hetzelfde minimum, en hoort volgens het criterium dus ook tot de optimale groeperingen. Toch is zij beter omdat de kleinste groep van de twee laatste groepen groter is dan die van de eerste groepering. De laatste groepering noemen we 'totaal-optimaal', omdat zij voldoet aan de eigenschap die hieronder wordt uiteengezet. Voor alle optimale groeperingen geldt dat de range minimaal is terwijl de kleinste groep maximaal is. De frequentieverdeling tussen de binnengrenzen van twee extreme groepen (dus exclusief deze twee), zijn verdeeld in zeg G_{12} groepen. Dit deel van de frequentieverdeling kan eveneens als een te optimaliseren G_{12} groeperingsprobleem worden beschouwd. Natuurlijk onder de voorwaarde dat $G_{12} > 1$. Hetzelfde geldt voor de twee frequentieverdelingen aan de buitenkant van de twee extreme groepen. Noemen we het oorspronkelijke groeperingsprobleem van orde 0 en de hoogstens drie daaropvolgende van orde 1, de hoogstens negen daaropvolgende van orde 2 enzovoort. Uiteindelijk wordt een orde d bereikt waarvoor geen enkele $G_{di} > 1$. Een groepering noemen we totaal-optimaal wanneer zij optimaal is voor iedere orde. Merk op dat er binnen een orde e meer dan 3^e deelproblemen kunnen optreden. In het bovenstaande voorbeeld zijn er in Groepering1 twee minimumgroepen van 61. In principe kunnen die twee maal drie orde 1 deelproblemen genereren. In het algemeen levert een deelprobleem van orde d maximaal $r_v \times r_x \times 3$ deelproblemen van orde $d + 1$, waarin r_v en r_x respectievelijk het aantal minimumgroepen en het aantal maximumgroepen in het deelprobleem van orde d . Zij $\lfloor x \rfloor$ het grootste gehele getal niet groter dan x . Bijvoorbeeld $\lfloor 1.2 \rfloor = \lfloor 1.0 \rfloor = 1$ (merk op dat $\lfloor -1.2 \rfloor = -2$, $\lfloor . \rfloor$ is dus niet hetzelfde als 'afknotten'). Over de maximale orde van een G -groepering hebben we de volgende

Stelling 1

De maximale orde van een G -groepering is $\lfloor G/2 \rfloor - 1$.

Bewijs

Het is eenvoudig in te zien dat een maximale orde van een G -groepering slechts dan gerealiseerd wordt wanneer de (unieke) extreme groepen voor iedere orde steeds aan de buitenkant van de i -de orde groepering worden gevonden. Zij kunnen zowel bij elkaar een van beide uitersten liggen of ieder aan een van de twee uitersten. Iedere orde genereert op deze manier slechts 1 hogere orde probleem in plaats van de maximale 3. Voor een 9-groepering krijgen we dan bijvoorbeeld 001122334 of 012343210, waarin de twee nullen de plaats aangeven van de twee extreme groepen van orde 0 enz.

De vierde orde bevat bij deze plaatsing van extreme groepen slechts een groep, en is derhalve niet meer te optimaliseren. Omdat iedere orde precies twee van de G groepen bevat heeft een G -probleem maximaal $\lfloor G/2 \rfloor$ ordes. Omdat de eerste orde rang 0 heeft is de maximale orde derhalve $\lfloor G/2 \rfloor - 1$. ■

Het is helaas niet het geval dat er voor iedere frequentie-verdeling en ieder aantal groepen precies één totaal-optimale groepering zou zijn, zoals het volgende zeer eenvoudige voorbeeld laat zien

2-Groepering1	1		2
Frequenties	1	1	1
2-Groepering2	2		1

Beide 2-groeperingen zijn totaal-optimaal, doch verschillend.

Naarmate een frequentieverdeling een grotere maximum score heeft en hogere frequenties bevat wordt het in het algemeen lastiger om een totaal-optimale of zelfs maar een optimale groepering te vinden. Noteer een frequentieverdeling van scores als een vector $\underline{f} = (f_0, f_1, \dots, f_s)$, en met P een willekeurige groepering van $\underline{f}$ in G groepen, soms met subscript P_G , om het aantal groepen van P aan te geven. De vector met het aantal observaties per groep wordt weergegeven met $\underline{n}_P = (n_{P1}, \dots, n_{PG})$ met $\sum_g n_{Pg} = N$, de score-bovengrenzen van de groepen met $\underline{c}'_P = (c_{P1}, \dots, c_{PG} = S)$, en de range van P met $r_P = \max(\underline{n}_P) - \min(\underline{n}_P)$. Omdat de score-bovengrens van de hoogste groep gelijk is aan de maximum score

$(c_{PG} = S)$, is een G -groepering van $\underline{f}$ bepaald door de eerste $G-1$ bovengrenzen $\underline{c} = c_{P1}, \dots, c_{PG-1}$. De verzameling van alle groeperingen van $\underline{f}$ in G groepen noteren we met Q_G , de verzameling van optimale G -groeperingen met O_G , en de verzameling van totaal-optimale groeperingen met T_G .

Het aantal elementen $|Q_G|$ van Q_G is gelijk aan

$$|Q_G| = \frac{(S+1)^{G-1}}{(G-1)!}.$$

Dit is gemakkelijk in te zien door te bedenken dat er $G-1$ scheidingsstreepjes moeten worden geplaatst op $S+1$ scores, waarbij op iedere score meer dan een scheidingsteken mag staan. Indien in Q_G geen lege groepen zijn toegestaan, en groeperingen met dezelfde n als equivalent worden beschouwd dan is

$$|Q_G| = \frac{(U-1)!}{(U-G)!(G-1)!},$$

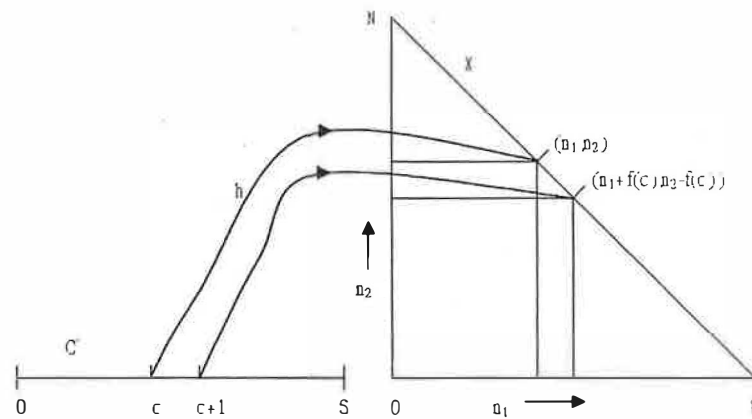
waarin U het aantal scores s , waarvoor $f_s > 0$. In beide gevallen is $|Q_G|$ veel te groot om uitputtend door te lopen op zoek naar een optimale G -groepering. Van het aantal elementen van O_G en T_G is minder gemakkelijk iets te zeggen.

Voor het beoordelen van de homogeniteit van een groepering is het van belang welke orde van grootte kan worden verwacht voor de range van een optimale groepering. Onderstaande stelling geeft een bovengrens voor de range van een optimale groepering.

Stelling 2

Zij f een frequentie-verdeling van scores $0..S$, met maximale frequentie $f^m = \max(f)$. Zij G een aantal scoregroepen, met $G \leq S$, en $P \in O_G$ een optimale groepering met $r_P = \min\{r_T : T \in Q_G\}$, dan is $r_P \leq f^m$.

Een summiere toelichting van het bewijs is te vinden in Figuur 1.



Figuur 1

Toelichting bij Stelling 2

Bewijs

Zij $V = \langle 0, \dots, S \rangle$ de geordende verzameling van scores, en zij C de verzameling van $(G-1)$ -dimensionele vectoren $\underline{c}$ waarvoor $c_i \in V$ en $0 \leq c_1 < c_2 < \dots < c_{G-1} < S$. Zij V' en C' de continue versies van V en C : V' het interval $[0, S]$ en C' de $(G-1)$ -dimensionele vectoren $\underline{c}'$, waarvoor $c'_i \in V'$ en $0 < c'_1 < c'_2 < \dots < c'_{G-1} < S$. Voor ieder element $\underline{c} \in C$, is er voor iedere i ($0 < i < G$) in beide richtingen een element $\underline{d} \in C$ met $|c_i - d_i| \leq 1$ en $c_j = d_j$ voor $j \neq i$, tenzij $\underline{c}$ op de rand ligt van C , in welk geval dit maar voor een richting geldt. Hieruit volgt dat raster C' door C bedekt wordt met punten, waarvan de burens componentsgewijs slechts 1 verschillen. Daarom is er bij ieder willekeurige vector $\underline{c}' \in C'$ altijd een naburig element $\underline{c} \in C$, waarvoor geldt dat $|c'_i - c_i| \leq 1/2$. Zij $N = \sum f_i$, en F de cumulatieve frequentieverdeling bij f . Zij c_0 gedefinieerd door $F(c_0) = 0$, en laat $c_G = S$, dus $F(c_G) = N$. Zij X' de $(G-1)$ -dimensionele deelverzameling van $\mathbb{R}^G$ met vectoren $\underline{x}'$, waarvoor geldt $\sum x_i = N$ en $x_i \geq 0$. Laat nu h de afbeelding zijn van C naar X' , met $h(\underline{c}) = F(c_1) - F(c_0)$, $F(c_2) - F(c_1), \dots, F(c_G) - F(c_{G-1}) = \underline{n}(\underline{c})$. Doordat een $\underline{c} \in C$ de bovengrenzen aangeeft van een groepering, correspondeert zij met precies 1 groepering $P \in Q_G$. De functie h beeldt $\underline{c}$ af op $\underline{n}$, de aantallen observaties per groep in P . Zij X de verzameling van alle vectoren $\underline{n}$ in het bereik van h . Nemen we aan dat $f_s > 0$ voor alle s , dan is het eenvoudig in te zien dat h een bijectie is van C naar X . Daarom wordt X' door het raster X op vergelijkbare manier bedekt als C' door C , met dit verschil dat de onderlinge componentsgewijze verschillen niet gelijk zijn aan 1 maar aan de elementen van f . Bovendien worden twee naburige elementen $\underline{c}$ en $\underline{c}'$ van C met $c_i - c'_i = 1$ en $c_j = c'_j$ voor $j \neq i$, door h afgebeeld op twee elementen $h(\underline{c}) = \underline{n}$ en $h(\underline{c}') = \underline{n}'$ van X , met

$$n_i - n'_i = n'_{i+1} - n_{i+1} = f_{c_i} \leq f^m,$$

die dus op twee achtereenvolgende componenten verschillen. Er blijft echter ook hier gelden dat er voor iedere willekeurige vector $\underline{x}' \in X'$ minstens een element $\underline{x}$ van X bestaat waarvoor geldt dat de componentsgewijze verschillen hoogstens gelijk zijn aan $f^m/2$: $|x'_i - x_i| \leq f^m/2$. Dit geldt in het bijzonder voor het element $\underline{y} = N/G, \dots, N/G$. Voor zo'n element van X in de buurt van $\underline{y}$ is de range maximaal gelijk aan f^m . ■

Afgezien van de zojuist afgeleide eigenschap ziet het raster X voor $G > 2$ er veel minder overzichtelijk uit dan C , en het is dan ook niet eenvoudig om een optimaal element uit X te vinden. Wanneer we als uitgangspunt een tamelijk homogene verdeling hebben kan een verbetering worden gevonden door kleine verschuivingen in de vector $\underline{c}$ (de score-bovengrenzen van de groepen) aan te brengen. Het is duidelijk dat dan alleen de frequenties die in de buurt van de bovengrenzen liggen een rol spelen. Wanneer het maximum van deze frequenties lager ligt dan het maximum van de gehele frequentieverdeling, kan de bovengrens voor de range van een optimale groepering zoals gegeven in de stelling overeenkomstig naar beneden worden bijgesteld. Het volgende corollarium geeft bovendien een beperking van de verzameling groeperingen waarin een (totaal-)optimale groepering zich zeker bevindt.

Corollarium

Voor een optimale G -groepering met groepsgroottes $\underline{n}$, en gemiddelde groepsgrootte $\bar{n} = N/G$ geldt voor $i = 1, \dots, G$,

$$\bar{n} - \frac{G-1}{G} f^m \leq n_i \leq \bar{n} + \frac{G-1}{G} f^m.$$

Bewijs

Zij $\underline{n} \in \mathbb{R}^G$, $\sum_i n_i = N$, $n_x = \max(\underline{n})$ en $n_v = \min(\underline{n})$. Merk op dat voor een $\underline{n}$ met een maximale n_x en maximale range $r = n_x - n_v = f^m$ moet gelden dat $n_i = n_v$ voor $i \neq x$. Zouden we immers voor een i n_i groter maken dan n_v dan moet n_x evenredig worden verlaagd, omdat $\sum n_i = N$. Dan is $n_x = N - (G-1)n_v = N - (G-1)(n_x - f^m)$. Voor een optimale $\underline{n}$ geldt dus dat

$$n_x \leq \bar{n} + \frac{G-1}{G} f^m.$$

Op analoge manier vinden we dat

$$n_v \leq \bar{n} - \frac{G-1}{G} f^m. \blacksquare$$

Van dit Corollarium wordt gebruik gemaakt in het later te behandelen algoritme *CSearch*.

Wanneer we willen dat de grenzen tussen groepen altijd samenvallen met de scores, zoals hierboven steeds gesteld, dan hebben we een deterministisch algoritme nodig voor het vinden van een groepering dicht bij het optimum. Staan we echter ook toe dat een score s met $F_{s-1} < Y_i$ en $F_s \geq Y_i$, voor een i , random wordt toegewezen aan groep i of $i + 1$, dan kan een maximale homogeniteit worden bereikt, zij het ten koste van een kleine achteruitgang van homogeniteit van de scores binnen groepen. Algoritmen met dit oogmerk noemen we random. Eerst worden enkele deterministische algoritmen behandeld, daarna wordt aandacht gegeven aan random algoritmen.

3 Deterministische algoritmen voor het vinden van een homogene groepering

Meestal zoekt men een groepering met een bepaald aantal groepen, bijvoorbeeld acht. Bovendien kan het gewenst zijn om de groeps grootte niet beneden een bepaald minimum te laten zakken. Beide wensen kunnen tegenstrijdig zijn, en in dat geval moet het aantal groepen worden verlaagd. In de hieronder besproken algoritmen wordt hiermee rekening gehouden. Van alle hier besproken algoritmen is in de Appendix een listing in pseudo-code gegeven, waar men een duidelijk beeld van de details kan krijgen. De eis van een minimaal aantal observaties per groep is hier echter uit weg gelaten. Er is een vast aantal groepen verondersteld.

Het eerste algoritme dat we bespreken, genaamd 'TIA', wordt als uitgangspunt genomen. Het is geïmplementeerd in het Toets- en Item-Analyse programma van het Cito genaamd 'TIA'. Het heeft als eigenschap dat bij $G > 3$ de bovengrens van de range van de opgeleverde groepering gelijk is aan $2 \times f^m$ dus twee maal zo groot als theoretisch mogelijk is (voor een bewijs zie hieronder). Daarna worden twee vervolgalgoritmen besproken, die de TIA-groepering verbeteren, en bijvoorbeeld de range zeker onder of op de bovengrens f^m brengen. Tenslotte worden twee algoritmen besproken, die een optimale groepering in een bepaalde zoekruimte rond $\underline{y}$ vinden. De range van de optimale groepering uit met name de eerste zoekruimte blijkt echter niet altijd beneden de theoretische bovengrens te liggen. Wanneer de resultaten van deze zoekalgoritmen ook worden onderworpen aan de zojuist genoemde vervolgalgoritmen wordt in het eerste geval meestal een betere oplossing gevonden. In ieder geval voldoet het resultaat van de vervolgalgoritmen wel altijd aan de theoretische bovengrens. Het optimum van het tweede zoekalgoritme is meestal niet meer te verbeteren door de vervolgalgoritmen. Met uitzondering van het tweede zoekalgoritme met een 'ruime' instelling kan in de andere gevallen niet worden gegarandeerd dat een groepering met minimale range wordt gevonden, laat staan een totaal-optimale groepering.

Het algoritme TIA

Stel dat de frequentie-verdeling f met cumulatieve frequentie-verdeling F , van in totaal N observaties, in G groepen moet worden gegroepeerd. Dan is de vector $\underline{n} = \underline{y}$ met range 0 gelijk aan

$$\underline{y} = N/G, N/G, \dots, N/G.$$

De vector $\underline{Y}$ met cumulatieve groep-groottes is derhalve

$$\underline{Y} = N/G, 2N/G, \dots, GN/G = N.$$

TIA bepaalt de vector $\underline{c}$ (de score-bovengrenzen van de groepen) waarvoor de cumulatieve verdeling F , zo dicht mogelijk bij $\underline{Y}$ komt:

$$|F_{c_i} - Y_i| = \min \{|F_c - Y_i| : c \in V\}, \text{ voor } i = 1, \dots, G-1.$$

Voor 'onbesliste' situaties raadplege men het algoritme in de Appendix. De theoretische bovengrens van de range van een TIA-groepering voor $G > 4$ is gelijk aan $2 \times f^m$. We bewijzen eerst een iets zwakkere stelling, en geven daarna de voorwaarden waaronder de genoemde bovengrens wordt gerealiseerd.

Stelling

Voor $G \geq 4$ bestaan er frequentieverdelingen f waarvoor de bovengrens van de range van een TIA-groepering gelijk is aan $2 \times f^m - 1$, dus bijna tweemaal hoger dan noodzakelijk is.

Bewijs

Zij $N \setminus G = \lfloor N/G \rfloor$, dan is eenvoudig in te zien dat een even maximale frequentie zodanig 'rond' ieder van de drie grenspunten 1, 2, en 3, geplaatst kan worden dat TIA

$$N_1 = N \setminus G + \frac{f^m}{2},$$

$$N_2 = 2N \setminus G - \frac{f^m}{2} + 1, \text{ en}$$

$$N_3 = 3N \setminus G + \frac{f^m}{2} \text{ oplevert.}$$

Hieruit volgt dat

$$n_2 = N \setminus G - f^m, \text{ en}$$

$$n_3 = N \setminus G + f^m - 1, \text{ zodat}$$

$$r \geq n_3 - n_2 = 2f^m - 1. \blacksquare$$

Voor $G \geq 5$, G oneven, f^m even, en $N = mG + 1$, voor gehele m , is zelfs een frequentieverdeling te construeren met TIA-range gelijk aan $2 \times f^m$. Leg hiervoor f^m maximaal 'ongelukkig' rond de scheidingspunten $G \setminus 2 - 1$, $G \setminus 2$ en $G \setminus 2 + 1$.

Bekijk als voorbeeld van zo'n TIA-'worst case' het volgende groeperingsprobleem:

$$G = 5, N = 20 \times 5 + 1 = 101, f^m = 6,$$

$$\underline{f} = 5, 5, 2, 6, 5, 5, 4, 6, 5, 5, 4, 6, \dots$$

Toepassing van TIA geeft $\underline{n} = 23, 14, 26, \dots$, dus $r = 26 - 14 = 12 = 2 \times f^m$.

Het volgende willekeurige groeperingsprobleem wordt verder ter toelichting van de algoritmen gebruikt:

Zij

$$\begin{array}{rcll} G & = & 4 & \\ N & = & 40 & \\ \underline{f} & = & 6 & 9 \quad 9 \quad 2 \quad 9 \quad 5, \text{ dus } f^m = 9 \\ \underline{F} & = & 6 & 15 \quad 24 \quad 26 \quad 35 \quad 40 \\ Y & = & 10 & 20 \quad 30 \quad 40 \end{array}$$

dan produceert TIA de groepering

$$\underline{c} = 0 \ 2 \ 3 \ 5 \text{ met } \underline{n} = 6 \ 18 \ 2 \ 14, \text{ en } \underline{N} = 6 \ 24 \ 26 \ 40.$$

Voor deze groepering is $r = 18 - 2 = 16 > 9$, en is derhalve zeker niet optimaal.

Vervolgalgoritme PairBalance

Gegeven een groepering P maakt PairBalance de aantallen observaties n_1 en n_2 in groepen 1 en 2 zoveel mogelijk aan elkaar gelijk door de bovengrens c_1 van groep 1 naar boven of naar onder te verleggen. Vervolgens hetzelfde met groepen 2 en 3, tot en met de groepen $G-1$ en G . Daarna doet PairBalance hetzelfde in omgekeerde richting, dus van groep $G-1$, en $G-2$ tot en met 2 en 1, net zo lang tot er geen verbetering meer optreedt.

Het voorbeeld, voortbouwend op het resultaat van TIA:

Verhogen c_1 van 0 naar 1 levert $n_1 = 15$, $n_2 = 9$, met $|n_1 - n_2| = 6 < 18 - 6 = 12$.

Verhogen c_3 van 3 naar 4 levert $n_3 = 11$, $n_4 = 5$, met $|n_3 - n_4| = 6 < 14 - 2 = 12$.

Deze veranderingen in $\underline{c}$ resulteren in

$\underline{c} = 1\ 2\ 4\ 5$, met $\underline{n} = 15\ 9\ 11\ 5$, en $\underline{N} = 15\ 24\ 35\ 40$.

Nu is $r = 15 - 5 = 10$, weliswaar verbeterd maar nog steeds groter dan $f^m = 9$, en dus zeker niet optimaal.

Vervolgalgoritme BelowMax

Zij P' een groepering van f . Bepaal hiervan de grootste en de kleinste groepen, respectievelijk x en v . Stelling 2 garandeert dat er een groepering P van het deel f' van f tussen de grenzen van x en v bestaat met $r_p \leq \max(f')$. Indien $n_x - n_v > \max(f')$ kan P worden verkregen door minstens een binnengrens, waarvan er minstens een van x of n , vanaf v in de richting van x te verschuiven. BelowMax begint met verschuiven bij v , omdat vergroting van n_v hogere prioriteit heeft dan verkleinen van n_x . Indien dit geen resultaat oplevert dan wordt begonnen bij het verkleinen van n_x en doorlopen in de richting van v . De binnengrens van de extreme groep wordt precies een scorepunt opgeschoven, en de grenzen van de daaropvolgende groepen l steeds zoveel als nodig is om het nieuwe aantal observaties n_l te laten voldoen aan $n_x > n_l > n_v$, waarin n_x en n_v de oorspronkelijke aantallen observaties in respectievelijk x en v . BelowMax wordt herhaald tot geen verbetering meer optreedt.

Het voorbeeld voortbouwend op het resultaat van PairBalance:

De extreme groepen zijn 1 en 4, met 4 de kleinste. De grenzen naar links opschuiven beginnend bij 4 levert: $\underline{c} = 0\ 1\ 3\ 5$, met $\underline{n} = 6\ 9\ 11\ 14$, en $\underline{N} = 6\ 15\ 26\ 40$.

Aan de waarden voor $\underline{n}$ kan men zien dat de range $r = 14 - 6 = 8 < 9$ in ieder geval kleiner is dan de maximum frequentie.

Zoekalgoritme BSearch

BSearch volgt een binaire zoekboom voor de eerste $G-1$ groepen. Nadat de grootte van eerste $G-1$ groepen is bepaald ligt de grootte van de laatste groep vast. Zij $\bar{n} = N/G$ de gemiddelde groeps-grootte. Van de verzameling Z met de 2^{G-1} vectoren $\underline{n}$ met n_i precies kleiner en precies groter dan $\bar{c}$ wordt de groepering met de kleinste range bepaald.

Nemen we weer het voorbeeld met

$$f = 6 \ 9 \ 9 \ 2 \ 9 \ 5, \ N = 40, \ G = 4. \text{ Voor } G - 1 = 3 \text{ is } |Z| = 2^3 = 8.$$

De onderstaande tabel geeft de verzameling Z en voor iedere groepering de range:

Groepering				Range
6	9	9	16	10
6	9	11	14	8 optimum
6	18	2	14	16
6	18	11	5	13
15	9	2	14	13
15	9	11	5	10
15	11	9	5	10
15	11	14	0	15

Merk op dat BSearch niet alle combinaties van precies onder en boven $\underline{Y}$ doorloopt, maar alle combinaties van $n_1, \dots, n_{G-1}$ onder en boven de $G-1$ dimensionele vector $\bar{n}, \dots, \bar{n}$.

Zoekalgoritme CSearch

CSearch maakt ook gebruik van Stelling 2 en het daaronder gegeven Corollarium waarin werd bewezen dat voor alle componenten van $\underline{n}$ van een optimale groepering geldt

$$\bar{n} - \frac{(G-1)}{G} f^m \leq n_i \leq \bar{n} + \frac{(G-1)}{G} f^m.$$

Hierbij moet worden opgemerkt dat de twee extremen in deze ongelijkheden een zeer kleine kans hebben om bij een willekeurige frequentieverdeling te worden gerealiseerd. In de hieronder te bespreken simulatie-studie bleek dat bij 8 groepen $d = 7/8 = 0.875$

vervangen kon worden door 0.6 ten koste van een nauwelijks merkbare verslechtering. Voor willekeurige G , zou derhalve (omdat $0.6 \times 8/7 = 0.7$) $d = 0.7 * (G - 1) / G$ een goed voorstel kunnen zijn. CSearch doorloopt alle vectoren $\underline{n}$ tussen deze grenzen, en vindt derhalve bij $d = (G - 1) / G$ zeker een optimale groepering. De zoekruimte, en daarmee de zoektijd, neemt echter wel aanzienlijk af door een iets kleinere d te kiezen. Overigens moet worden opgemerkt dat de zoektijd per probleem zo kort is, dat zij voor geen enkel van de hier besproken algoritmen betrouwbaar te meten was. Wel is duidelijk dat CSearch de langste tijd vergt, maar die moet dan rond de 0.001 seconde liggen (met een 486-100 systeem onder DOS 6.1). Voor een 'gewone' min of meer eentoppige frequentieverdeling met scores 0...250, 900 observaties en 9 groepen was nog steeds minder dan 1 seconde nodig.

Gegeven een optimale groepering kan natuurlijk een totaal-optimale groepering worden verkregen door CSearch toe te passen op de hogere orde deelfrequentieverdelingen. Dit zou echter onnodige extra zoektijd vergen, omdat de totaal-optimale groepering reeds tijdens het zoeken van een optimale groepering op optimaliteit is gecontroleerd. Het is waarschijnlijk efficiënter om iedere optimale groepering op totaal-optimaliteit te testen.

4 Een simulatiestudie

Voor 8 verschillende maximale frequenties werd 1000 keer een 'willekeurige' frequentie-verdeling gegenereerd voor scores van 0 tot 100. Voor een bepaalde maximale frequentie f^m werd er voor iedere score een random getal getrokken tussen 0 en f^m . Er zijn 8 niveaus voor f^m gekozen: $f^m = 2^n - 1$, voor $n = 2, \dots, 9$. Voor ieder van deze frequentieverdelingen is door ieder van de drie algoritmen (TIA, BSearch, en CSearch) een groepering berekend. Ieder van de drie algoritmen werd gevolgd door PairBalance (P), BelowMax (M), en nogmaals PairBalance. Voor TIA wordt voor ieder van de achtereenvolgende vervolgalgoritmen apart gerapporteerd (P, PM en PMP), omdat die ieder nog een aanzienlijke verbetering geven. Voor de andere twee algoritmen wordt alleen het resultaat van het geheel van de vervolgalgoritmen (PMP) gerapporteerd. In de simulatiestudie is alleen naar optimaliteit gezocht, totaal-optimaliteit is genegeerd.

Per maximale frequentie worden twee tabellen gerapporteerd die de prestaties van de algoritmen over 1000 simulaties samenvatten. De eerste tabel bevat voor ieder van de algoritmen de volgende prestatiematen:

MeanRng	: de gemiddelde range van de groeperingen
StdvRng	: de standaarddeviatie van de range van de groeperingen
MinR	: de minimale range
MaxR	: de maximale range
%MinRng	: het percentage keren dat het betreffende algoritme een groepering vond voor een bepaalde frequentieverdeling met de kleinste range van de algoritmen

De eerste regel van iedere tabel bevat enkele karakteristieken van de simulatie:

FreqSz	: de maximale frequentie
MGrpSz	: het gemiddelde aantal observaties per groep.
M#Grps	: het gemiddelde aantal groepen. Dit is afhankelijk van de maximale frequentie, het minimum aantal observaties per groep (50), en het maximum aantal groepen (8).
M#Obs	: het gemiddelde aantal observaties per simulatie.

Ieder van de volgende acht tabellen is het resultaat van 1000 simulaties met random frequentieverdelingen voor scores 0,...,100, en mogelijke maximale frequentie $\text{MaxFr} = 2^{\#Tabel + 1} - 1$.

Tabel 1

MaxFr	M#Grps	MGrpSz	M#Obs						
3	2.7	59	158.2						
	Method	MeanRng	StdvRng	MinR	MaxR	%MinRng			
	TIA	2.30	1.41	0	9	92.7			
	TIA&P	2.21	1.28	0	7	99.5			
	TIA&PM	2.21	1.28	0	7	99.5			
	TIA&PMP	2.21	1.28	0	7	99.5			
	B	2.21	1.28	0	7	99.5			
	B&PMP	2.21	1.27	0	7	100.0			
	C	2.21	1.28	0	7	99.9			
	C&PMP	2.21	1.27	0	7	100.0			

#Times row method had lower Range than column method										Total
		1	2	3	4	5	6	7	8	
1.	TIA	0	0	0	0	1	0	1	0	2
2.	TIA&P	68	0	0	0	2	0	1	0	71
3.	TIA&PM	68	0	0	0	2	0	1	0	71
4.	TIA&PMP	68	0	0	0	2	0	1	0	71
5.	B	69	2	2	2	0	0	1	0	76
6.	B&PMP	73	5	5	5	5	0	1	0	94
7.	C	73	5	5	5	5	0	0	0	93
8.	C&PMP	73	5	5	5	5	0	1	0	94
Total		492	17	17	17	22	0	7	0	572

Tabel 2

MaxFr	M#Grps	MGrpSz	M#Obs						
7	6.2	58	357.7						
	Method	MeanRng	StdvRng	MinR	MaxR	%MinRng			
	TIA	7.71	2.89	2	18	44.3			
	TIA&P	6.63	2.31	2	14	77.2			
	TIA&PM	6.51	2.23	2	14	82.4			
	TIA&PMP	6.49	2.23	2	14	83.2			
	B	6.32	2.01	2	13	88.5			
	B&PMP	6.21	1.94	2	13	97.9			
	C	6.17	1.92	2	13	100.0			
	C&PMP	6.17	1.92	2	13	100.0			

#Times row method had lower Range than column method										Total
		1	2	3	4	5	6	7	8	
1.	TIA	0	0	0	0	17	0	0	0	17
2.	TIA&P	426	0	0	0	69	1	0	0	496
3.	TIA&PM	451	63	0	0	72	1	0	0	587
4.	TIA&PMP	451	63	8	0	73	1	0	0	596
5.	B	515	195	143	137	0	0	0	0	990
6.	B&PMP	545	211	158	150	96	0	0	0	1160
7.	C	557	228	176	168	115	21	0	0	1265
8.	C&PMP	557	228	176	168	115	21	0	0	1265
Total		3502	988	661	623	557	45	0	0	6376

Tabel 3

MaxFr	M#Grps	MGrpSz	M#Obs						
15	8.0	95	756.3						
	Method	MeanRng	StdvRng	MinR	MaxR	%MinRng			
	TIA	18.80	6.47	5	47	21.3			
	TIA&P	15.51	4.63	5	33	52.9			
	TIA&PM	14.95	4.27	5	28	59.6			
	TIA&PMP	14.85	4.26	5	28	62.4			
	B	14.09	3.61	5	26	78.6			
	B&PMP	13.80	3.53	5	25	92.5			
	C	13.67	3.41	5	24	100.0			
	C&PMP	13.67	3.41	5	24	100.0			

#Times row method had lower Range than column method

		1	2	3	4	5	6	7	8	Total
1.	TIA	0	0	0	0	17	3	0	0	17
2.	TIA&P	598	0	0	0	65	8	0	0	671
3.	TIA&PM	636	135	0	0	74	10	0	0	855
4.	TIA&PMP	636	135	35	0	81	11	0	0	898
5.	B	742	404	335	310	0	0	0	0	1791
6.	B&PMP	762	423	357	328	149	0	0	0	2019
7.	C	787	471	404	376	214	75	0	0	2327
8.	C&PMP	787	471	404	376	214	75	0	0	2327
Total		4948	2039	1535	1390	881	182	0	0	10905

Tabel 4

MaxFr	M#Grps	MGrpSz	M#Obs						
31	8.0	197	1573.9						
	Method	MeanRng	StdvRng	MinR	MaxR	%MinRng			
	TIA	37.50	12.58	9	78	20.4			
	TIA&P	31.02	9.43	9	59	53.7			
	TIA&PM	30.27	8.89	9	56	58.6			
	TIA&PMP	30.09	8.88	9	56	61.3			
	B	28.71	7.74	9	52	72.3			
	B&PMP	28.16	7.54	9	51	87.8			
	C	27.81	7.26	9	49	100.0			
	C&PMP	27.81	7.27	9	49	100.0			

#Times row method had lower Range than column method

		1	2	3	4	5	6	7	8	Total
1.	TIA	0	0	0	0	12	1	0	0	13
2.	TIA&P	608	0	0	0	94	11	0	0	713
3.	TIA&PM	650	111	0	0	108	13	0	0	882
4.	TIA&PMP	650	111	31	0	118	15	0	0	925
5.	B	741	383	323	301	0	0	0	0	1748
6.	B&PMP	763	399	341	314	164	0	0	0	1981
7.	C	796	463	414	387	277	122	0	0	2459
8.	C&PMP	796	463	414	387	277	122	0	0	2459
Total		5004	1930	1523	1389	1050	284	0	0	11180

Tabel 5

MaxFr	M#Grps	MGrpSz	M#Obs							
63	8.0	397	3177.7							
	Method	MeanRng	StdvRng	MinR	MaxR	%MinRng				
	TIA	75.13	26.22	19	177	20.1				
	TIA&P	62.65	18.68	19	123	49.8				
	TIA&PM	60.76	17.48	19	109	54.9				
	TIA&PMP	60.41	17.42	19	109	57.9				
	B	57.47	14.74	19	106	70.0				
	B&PMP	56.29	14.38	19	104	88.1				
	C	55.58	13.79	19	102	100.0				
	C&PMP	55.58	13.79	19	102	100.0				
#Times row method had lower Range than column method										
		1	2	3	4	5	6	7	8	Total
1.	TIA	0	0	0	0	17	1	0	0	18
2.	TIA&P	598	0	0	0	98	11	0	0	707
3.	TIA&PM	638	121	0	0	108	11	0	0	878
4.	TIA&PMP	638	121	42	0	123	14	0	0	938
5.	B	752	421	356	329	0	0	0	0	1858
6.	B&PMP	769	440	382	349	198	0	0	0	2138
7.	C	799	502	451	421	300	119	0	0	2592
8.	C&PMP	799	502	451	421	300	119	0	0	2592
Total		4993	2107	1682	1520	1144	275	0	0	11721

Tabel 6

MaxFr	M#Grps	MGrpSz	M#Obs							
127	8.0	800	6402.3							
	Method	MeanRng	StdvRng	MinR	MaxR	%MinRng				
	TIA	153.09	52.34	29	335	19.4				
	TIA&P	127.29	39.74	27	242	51.4				
	TIA&PM	123.67	37.17	27	221	54.4				
	TIA&PMP	122.77	36.93	27	221	58.2				
	B	117.10	31.81	27	213	68.1				
	B&PMP	114.42	30.96	27	213	87.3				
	C	112.71	29.47	27	187	100.0				
	C&PMP	112.71	29.47	27	187	100.0				
#Times row method had lower Range than column method										
		1	2	3	4	5	6	7	8	Total
1.	TIA	0	0	0	0	19	3	0	0	22
2.	TIA&P	630	0	0	0	116	13	0	0	759
3.	TIA&PM	663	127	0	0	132	16	0	0	938
4.	TIA&PMP	663	127	50	0	148	16	0	0	1004
5.	B	755	401	352	324	0	0	0	0	1832
6.	B&PMP	779	420	376	338	208	0	0	0	2121
7.	C	806	486	456	418	319	127	0	0	2612
8.	C&PMP	806	486	456	418	319	127	0	0	2612
Total		5102	2047	1690	1498	1261	302	0	0	11900

Tabel 7

MaxFr	M#Grps	MGrpSz	M#Obs						
255	8.0	1615	12917.0						
	Method	MeanRng	StdvRng	MinR	MaxR	%MinRng			
	TIA	300.21	102.54	64	651	19.8			
	TIA&P	252.79	76.75	64	556	47.3			
	TIA&PM	244.56	70.47	64	472	52.2			
	TIA&PMP	243.12	70.62	64	472	55.2			
	B	229.83	59.87	64	425	70.8			
	B&PMP	225.54	59.08	64	425	88.6			
	C	222.90	56.66	64	379	100.0			
	C&PMP	222.90	56.66	64	379	100.0			

#Times row method had lower Range than column method

		1	2	3	4	5	6	7	8	Total
1.	TIA	0	0	0	0	16	1	0	0	17
2.	TIA&P	607	0	0	0	92	6	0	0	705
3.	TIA&PM	634	135	0	0	98	6	0	0	873
4.	TIA&PMP	634	135	38	0	115	6	0	0	928
5.	B	750	457	385	362	0	0	0	0	1954
6.	B&PMP	772	464	398	368	194	0	0	0	2196
7.	C	802	527	478	448	292	114	0	0	2661
8.	C&PMP	802	527	478	448	292	114	0	0	2661
Total		5001	2245	1777	1626	1099	247	0	0	11995

Tabel 8

MaxFr	M#Grps	MGrpSz	M#Obs						
511	8.0	3226	25808.0						
	Method	MeanRng	StdvRng	MinR	MaxR	%MinRng			
	TIA	609.30	202.70	176	1348	16.5			
	TIA&P	505.81	154.70	162	988	46.1			
	TIA&PM	491.51	142.96	156	931	50.8			
	TIA&PMP	489.12	142.43	156	931	53.0			
	B	461.17	123.23	156	871	68.6			
	B&PMP	452.17	120.19	156	871	85.9			
	C	446.21	115.07	156	763	100.0			
	C&PMP	446.21	115.07	156	763	100.0			

#Times row method had lower Range than column method

		1	2	3	4	5	6	7	8	Total
1.	TIA	0	0	0	0	25	2	0	0	27
2.	TIA&P	621	0	0	0	103	11	0	0	735
3.	TIA&PM	657	122	0	0	112	12	0	0	903
4.	TIA&PMP	657	122	37	0	124	12	0	0	952
5.	B	775	447	385	369	0	0	0	0	1976
6.	B&PMP	797	459	405	381	201	0	0	0	2243
7.	C	835	539	492	470	314	141	0	0	2791
8.	C&PMP	835	539	492	470	314	141	0	0	2791
Total		5177	2228	1811	1690	1193	319	0	0	12418

In de eerste deeltabel van tabel 4 lezen we dat de gemiddelde range van de 1000 ranges gelijk is aan 37.50 en de standaarddeviatie 12.58. Het minimum van de 1000 ranges van TIA is 9, en het maximum 78. Verder had TIA in 20.4% van de 1000 dus 204 keer een range die minstens zo laag was als de ranges van de overige zeven zoekmethoden.

De tweede tabel is een kruistabel en bevat in iedere cel het aantal keren dat het rij-algoritme een lagere range voor een bepaalde frequentie-verdeling bereikte dan het kolom-algoritme. Bijvoorbeeld in de eerste regel van de tweede deeltabel van tabel 4 lezen we dat TIA 12 keer een lagere range vond dan BSearch en 1 keer een lagere dan BSearch gevolgd door PMP. De overige routines werden 0 keer door TIA verbeterd. In kolom 7 en 8 zien we dat CSearch, niet of wel gevolgd door PMP, geen enkele keer door de overige routines is verbeterd. Dit betekent tevens dat de PMP vervolgroutines geen range-verbetering realiseerden in de groepering die CSearch als optimum aanwees.

Bestudering van tabellen 1 tot en met 8 laat zien dat er ten opzichte van het algoritme TIA aanzienlijke winst valt te behalen. Dit wordt niet alleen duidelijk aan de lagere gemiddelde range die met de overige algoritmen wordt bereikt (de gemiddelde range van CSearch is ongeveer $3/4$ van het gemiddelde van TIA), maar met name ook aan de maximale range. De slechtste prestatie (MaxR) van het beste algoritme (CSearch) is ongeveer $3/5$ van de MaxR van TIA. Verder wordt de prestatie van TIA van tabel 3 in meer dan 78% van de simulaties overtroffen door CSearch (tweede deeltabel entry (7,1)), terwijl dit laatste algoritme geen enkele keer wordt overtroffen.

5 Een Random algoritme

Indien de aantallen observaties per groep gelijk moeten zijn, moet een random algoritme worden gebruikt. Het volgende random algoritme zorgt ervoor dat de aantallen observaties per groep hoogstens 1 verschillen. Indien component i van $\mathcal{Y}$ samenvalt met component j van $\mathcal{E}$, dan is er voor de bovengrens van groep i geen probleem. Alle scores lager of gelijk aan j worden ingedeeld in groep i of lager. Wanneer echter $F_{j-1} < Y_i < F_j$, dan kunnen de observaties met score j random worden ingedeeld in groep i of $i + 1$, met respectievelijk proporties

$$p_{i+1|j} = \frac{F_j - Y_i}{f_j} \text{ en } p_{i|j} = 1 - p_{i+1|j}.$$

Tijdens het lezen van een bestand kan men deze proporties exact verkrijgen door

$$p_{i+1|j} = \max \left[0, \frac{F_j - Y_i - r_{i+1|j}}{f_j - (r_{i|j} + r_{i+1|j})} \right],$$

waarin $r_{i|j}$ het aantal records met score j dat tot nu toe bij groep i is ingedeeld.

6 Conclusie

In dit rapport is het groeperen van frequentieverdelingen in groepen van zoveel mogelijk gelijke grootte besproken. Er is een bovengrens afgeleid voor het maximale verschil tussen de grootste en de kleinste groep van een optimale groepering. Verder zijn enkele methoden besproken om een homogene groepering te vinden. Het best presterende algoritme is CSearch dat zo kan worden ingesteld dat altijd een optimale groepering wordt gevonden. Het slechts presteert TIA. De theoretische bovengrens van de range van een door TIA gevonden groepering is ongeveer twee maal zo groot als nodig is. In de praktijk is het verschil in prestaties tussen CSearch en TIA minder groot. De range van een TIA-groepering is gemiddeld $4/3$ keer groter dan het optimum dat door CSearch is gevonden. In de slechtste gevallen (MaxR) is dit $5/3$. Het random algoritme deelt een frequentieverdeling zeker groepgroottes in die maximaal 1 observatie verschillen.

Notatie

Toelaten van $f_s=0$ introduceert enkele complicaties, die het begrip van de routines onnodig bemoeilijken. Ondanks dat deze pseudo-routines zorgvuldig zijn geschreven en nagelezen kunnen ze niet worden getest. Waarschijnlijk zullen ze daarom nog meer fouten bevatten dan gewone broncode. De PASCAL-broncode (Borland 6.0) is te verkrijgen bij de auteur.

27

```

PROCEDURE PairBalance
BEGIN
  REPEAT
    ready:=TRUE
    FOR g:=G-1 TO 1 DO BEGIN
      t:=cg
      j:=ft
      WHILE (cg>0) AND
        (min(ng+1+j, ng-j) > min(ng+1, ng)) DO BEGIN
        ready:=FALSE
        t:-1
        cg:=t
        j:=ft
      END
    END
    FOR g:=1 TO G-1 DO BEGIN
      t:=cg+1
      j:=ft
      WHILE (t<S) AND
        (min(ng+1-j, ng+j) > min(ng+1, ng)) DO BEGIN
        ready:=FALSE
        cg:=t
        t:+1
        j:=ft
      END
    END
  UNTIL ready
END { PairBalance }
=====

```

PROCEDURE BelowMax

De vier Routines met namen als FromVtoXRight voeren het omhoog- of omlaagschuiven van c_g uit in de aangegeven richting van v naar x of omgekeerd. Bijvoorbeeld:

```

PROCEDURE FromVtoXRight
{
  zij  $n_v=n_v$  en  $n_x=n_x$ 
  begin bij  $g=v$ ,  $c_v:=+1$ , en verhoog vervolgens indien nodig
 $c_g$ 
  voor de groepen g tot aan x totdat  $n_v<n_g<n_x$ .
  Stop bij x of bij de eerste groep die geen verandering
  behoeft
}
BEGIN { FromVtoXRight }
  ms:= $n_v+1$ 
  g:=v
   $c_v:=+1$ 
  REPEAT
    g:=+1
    ready:=TRUE
    WHILE  $n_g<ms$  DO BEGIN
      ready:=FALSE
       $c_g:=+1$ 
    END
  UNTIL ready or g=x
END { FromVtoXRight }

BEGIN { BelowMax }
  IF r=0 THEN EXIT
  minr:=r
  REPEAT
    oldr:=minr
    cc:=c
    IF v<x THEN BEGIN
      FromVtoXRight
      IF r>=oldr THEN BEGIN
        c:=cc
        FromXtoVLeft
      END
    END ELSE BEGIN
      FromVtoXLeft
      IF r>=oldr THEN BEGIN
        c:=cc
        FromXtoVRight
      END
    END
    minr:=r
  UNTIL minr>=oldr
END { BelowMax }
=====

```

```
PROCEDURE BSearch
```

```
  PROCEDURE groepeer
```

```
    FUNCTION bin(w,g) : {TRUE,FALSE}
    BEGIN
      bin:=TRUE als de g-de bit van w = 0, anders
      bin:=FALSE
    END
```

```
  BEGIN {groepeer}
    s:=0
    FOR g:=1 to G-1 DO BEGIN
      under:=bin(nxt,g)
      l:=fs
      WHILE g<G AND s<S DO BEGIN
        IF l>0 THEN BEGIN
          IF under THEN s:=-1
          cg:=s
          l:=0
          g:=+1
          under:=bin(nxt,g)
        END
        s:=+1
        l:=+fs
      END
    END
  END {groepeer}
```

```
  BEGIN {BSearch}
    minr:=N {groot getal}
    nxt:=0
    FOR ii:=1 TO 2G-1 DO BEGIN
      { loop over alle ng (g<G) precies onder en
      boven 0 }
      Groepeer { vind de groepering behorend bij nxt }
      IF r<minr THEN BEGIN { behoud de groepering met laagste
      r }
        minr:=r
        cc:=c
      END
      nxt:=+1
    END
    c:=cc { de groepering met laagste r }
  END { BSearch }
  =====
```

```

PROCEDURE CSearch
CONST beperk:=0.7 {  $0 < \text{beperk} \leq 1$  }

PROCEDURE reset(l) {Als  $n_g > \text{maxn}$  reset  $\underline{c}$  tot en met groep l }
BEGIN
  FOR g:=G to l DO
    WHILE  $n_g > \text{maxn}$  DO  $c_{g-1} := +1$  {  $n_g$  verandert mee met  $c_g!$  }
  END { reset }

PROCEDURE Nextc {vind de volgende groepering binnen de
grenzen} BEGIN
  ok := FALSE
  g:=G-1
  REPEAT { zoek een groep die kan worden opgehoogd }
    k:= $n_g$ 
    s:= $c_g$ 
    bfound:= $s < S$ 
    IF bfound THEN BEGIN
      s:=s+1
      k:=k+fs
      bfound:= $k \leq \text{maxn}$ 
    END
    IF bfound THEN BEGIN {zet de hoger geïndiceerde groepen }
      { precies boven het minimum }
       $c_g := k$ 
      FOR h:=g+1 TO G-1 DO BEGIN
        k:=0
        WHILE ( $k < \text{minn}$ ) AND ( $s < S$ ) DO BEGIN
          s:=s+1
          k1:=k+fs
        END
         $c_h := s$ 
      END
      ok:=  $n_g \geq \text{minn}$ 
      IF  $n_g > \text{maxn}$  THEN reset(g+2)
    END
  UNTIL bfound OR cdone
END { NextC }

```

```

BEGIN { CSearch }
  minr:=N { groot getal }
  a:=beperk*(G-1)/G
  b:=a*fm
  minn:=o-b { de tussengrenzen voor ng }
  maxn:=o+b
{ initiële groepering n1...nG-1 precies boven de ondergrens }
  s:=0
  k:=fs
  FOR g:=1 TO G-1 DO BEGIN
    WHILE k<minn DO BEGIN
      s:=+1
      k:=+fs
    END
    Cg:=s
    k:=0
  END
  reset(2)
  ok:=TRUE
{ loop over alle groeperingen binnen de grenzen }
  REPEAT
    IF OK THEN BEGIN
      IF r<minr THEN BEGIN
        cdone:=r=0
        minr:=r
        CC:=C
      END
    END
    nextc
  UNTIL cdone
  C:=CC
END { CSearch }
=====

```


Reeds eerder in 1998 verschenen OPD Memoranda

- 98-1 A.P.J.M. Heuvelmans. Constructie en verwerking van vragenlijsten.
- 98-2 A.P.J.M. Heuvelmans en J. Kuijt. Grafische afbeeldingen voor Computer Based Toetsen.

